

ニューラルネットによる関数近似について

成 久 洋 之・福 田 典 昭*

岡山理科大学工学部情報工学科

*岡山理科大学大学院工学研究科修士課程情報工学専攻

(1996年10月7日 受理)

1. はじめに

人工知能の一分野として広く応用されているニューラルネットワーク(Neural Network)における学習, 認識, 分析は入力データと出力データとの写像関係を決定しているものであると定義づけられる。すなわち, 学習データを用いて入力データに対する近似出力関数の精度を向上させるアルゴリズムと見なすことができる。

一般に多層ニューラルネットワークの隠れ層の出力関数としては, 超平面による線形分離を行うシグモイド関数(Sigmoid Function)が広く用いられているが, これとは異なる性質の超楕円による分離を行う放射状基底関数(Radial Basic Function: RBF)も現在盛んに研究されている。クラスタリング問題においては, シグモイド関数は大域的な分離をわずかな素子数で実現することができ, 微分形が簡単なためバックプロパゲーション法(Back Propagation: BP)による学習に適している。一方, RBF は一つの素子の決定領域が超楕円で表されることからパターン空間内でのクラスタの中心位置, 広がり的情報を容易に得ることができる。しかし, 領域面が超平面である場合は, その分類精度は悪くなり, 空間全域をカバーするために多数の素子を必要とする。

以上のように, 各関数は相反的な性質を持ち, 明らかに得意な対象が異なってくる。関数近似に限定すると, ニューラルネットとしては一般的に使用されているシグモイド関数よりは RBF の方がより効果的であるとされている。

本研究では, 関数近似において, RBF としてガウス関数を使用したニューラルネットと, 一般的に使用されている BP ニューラルネットとを比較し, その特性, 有効性を検討するものである。

2. バックプロパゲーション (BP) について

教師あり学習の一つである BP ネットワークは, 文字認識, 画像分類, 信号符号化等の様々なパターン解析問題に適用されてきた。このネットワークは, 大抵幾つかの層を持っており, 上下の層と完全結合している。ネットワークに入力が与えられると, 活性値の更新は, 処理ユニットの入力層から各内部層を通り出力層へと伝搬し, 出力ユニットがネッ

トワークの応答を答える。ネットワークがその内部パラメータを訂正するときは、その訂正機構は出力ユニットに始まって隠れ層を通り、入力層へと逆伝搬していく。従って、BP は誤差逆伝搬法とも呼ばれる。

2.1 BP の学習則

BP の学習則は、デルタ則を拡張した一般化デルタ則を用いる。デルタ則は、出力層と隠れ層間のシナプス荷重に関してのみ、その修正式を与える。これを出力層以前のすべての層間のシナプス荷重の修正に拡張したのが一般化デルタ則である。BP の学習則は以下の通りである。

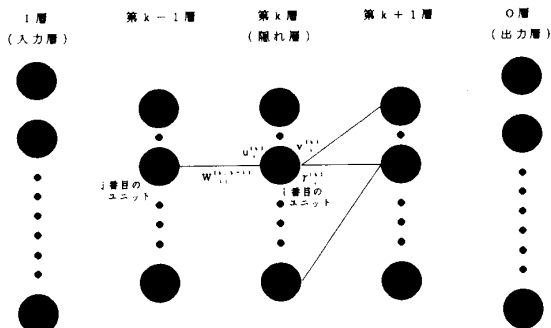


図1 階層的神経回路網の構成

図1のように、 K 層 (I 層を入力層とする) に属する i 番目のニューロンの入力・出力信号を、それぞれ $u_i^{(k)}$, $v_i^{(k)}$ とすると、

$$u_i^{(k)} = \sum_{j=1}^{N_{k-1}} W_{ij}^{(k,k-1)} v_j^{(k-1)} + \theta_i^{(k)} \quad (1)$$

$$v_i^{(k)} = f(u_i^{(k)}) \quad (2)$$

で与えられる。ここで $W_{ij}^{(k,k-1)}$ は $(k-1)$ 層内の j 番目のニューロンと k 層内の i 番目のニューロンのシナプス荷重、 N_{k-1} は $(k-1)$ 層に含まれるニューロン数である。また、 $\theta_i^{(k)}$ は k 層内の i 番目のニューロンのしきい値、関数 f はニューロンの入出力関数で、通常シグモイド関数が用いられる。学習は、教師信号に対するネットワークの出力信号の二乗誤差として定義される評価関数が最小値に収束するように実行される。これを各層間に適用して、 k 層における望ましい出力 (出力層であれば教師信号) を $v_i'^{(k)}$ ($i = 1, 2, \dots, N_k$)、実際に得られる出力を $v_i^{(k)}$ ($i = 1, 2, \dots, N_k$) とすれば、評価関数 E は、

$$E = \frac{1}{2} \sum (v_i'^{(k)} - v_i^{(k)})^2 \quad (3)$$

で与えられる。この E を最小にするように、ネットワークのシナプス荷重を変化させる。

学習則を求めた結果を示せば、シナプス荷重の修正量 $\Delta W_{ij}^{(k,k-1)}$ は、

$$\Delta W_{ij}^{(k,k-1)} = \eta \gamma_i^{(k)} v_j^{(k,k-1)} \quad (4)$$

で与えられる。ここで、 η は収束の速さを決める学習パラメータである。 $\gamma_i^{(k)}$ は k 層が出力層 (O 層) の場合には、

$$\gamma_i^{(k)} = (d_i - v_i^{(0)}) f'(u_i^{(0)}) \quad (5)$$

である。ここで、 $d_i (i = 1, 2, \dots, N_{(0)})$ は教師信号である。一方、 k 層が O 層以外のときは、

$$\gamma_i^{(k)} = \left(\sum_j W_{ij}^{(k+1,k)} \gamma_j^{(k-1)} \right) f(u_i^{(k)}) \quad (6)$$

である。式 (4)~(6) より、任意の k 層の修正信号 $\Delta W_{ij}^{(k,k-1)}$ は、1 層前の修正信号 $\Delta W_{ij}^{(k+1,k)}$ から再帰的に計算されることが分かる。

2.2 入出力関数（シグモイド関数）

BP ネットワークにおいて、入出力関数は導関数を持つ関数でなければならない。その点において、シグモイド関数は増加関数で、しかも連続な非線形関数であるため最も有効である。シグモイド関数は、次式のように表される関数である。

$$f(x) = 1/[1 + \exp(-x)] \quad (7)$$

シグモイド関数の一次導関数 $df(x)/dx$ は、以下の様に自らの関数 $f(x)$ で一次導関数を表すことができる関数である。

$$\frac{df(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{(1/f(x))-1}{(1/f(x))^2} = f(x)(1-f(x)) \quad (8)$$

従って、シグモイド関数を入出力関数として用いた場合、シナプス荷重の修正量を求めるための γ は次式のように表される。

$$\begin{aligned} \gamma_i^{(0)} &= (d_i - v_i^{(0)}) f(u_i^{(0)}) (1 - f(u_i^{(0)})) \\ &= (d_i - v_i^{(0)}) v_i^{(0)} (1 - v_i^{(0)}) \end{aligned} \quad (9)$$

$$\begin{aligned} \gamma_i^{(k)} &= \sum_j (W_{ij}^{(k+1,k)} \gamma_j^{(k-1)}) f(u_i^{(k)}) (1 - f(u_i^{(k)})) \\ &= \sum_j (W_{ij}^{(k+1,k)} \gamma_j^{(k-1)}) v_i^{(k)} (1 - v_i^{(k)}) \end{aligned} \quad (10)$$

$$(k = 2, 3, \dots, n-1)$$

上記の (9) 式は出力層のユニットに対して、(10) 式はそれ以外の層のユニットに対しての荷重の修正量を求める場合において用いられる。

3. 放射状基底関数（RBF）について

多重パーセプトロンの派生技術として、RBF として知られる関数を使用する方法がある。この関数はガウス関数を基本とするもので、パターン空間をうまく分割するような非線形関数である。RBF を用いたアプローチでは、パターン空間を分類するための超楕円を使用する。次項に、単一出力、多重出力 RBF ネットの構成図を示す。

図 2, 3 における x は入力ベクトルを示しており、隠れ層ユニット i の出力 h_i は、

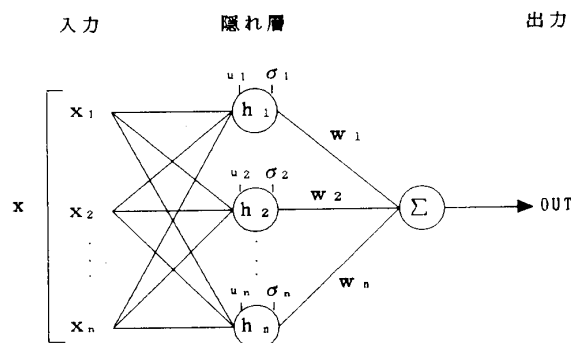


図2 単一出力 RBF ネット

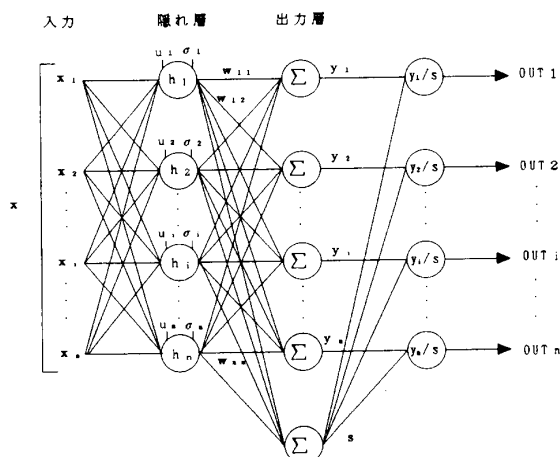


図3 多重出力 RBF ネット

$$h_i = \exp[-D_i^2/(2\sigma^2)] \quad (11)$$

$$D_i^2 = (x - u_i)^T(x - u_i) \quad (12)$$

によって表すことができる。ここで、 σ は有意領域の制御定数であり、標準偏差を意味する。また、 u_i は隠れ層ユニット i における荷重ベクトルを表している。

$$out = \sum_i h_i w_i \quad (13)$$

$$out_j = \sum_i h_i w_{ij} / \sum_i h_i \quad (14)$$

また、上記の (13), (14) 式によって、各々単一出力、多重出力 RBF ネットの出力値が得られる。(14) 式における多重出力 RBF ネットの出力値は、全ての隠れ層ユニットの出力総和が一定になるように正規化操作を行う処理が必要となる。

以上のように、ガウス関数を用いることによって各隠れ層ユニットの非線形特性を実現している。そのため、このネットは確率的分類器 (Probabilistic Neural Net Classifier) の範疇に属していると言える。

3.1 回帰ニューラルネット (GRNN) について

先に記した RBF を用いた方法に、一般化された回帰ニューラルネットワーク (Generalized Regression Neural Network: GRNN) がある。このネットは、シグモイド関数を用いた BP の様な繰り返し学習の必要性が全くない望ましい特質を持つ。GRNN は本質的には RBF ネットであり、ベイズ分類手法に基づいた関数近似に適しているネットであると言える。次項に GRNN の構成図を示す。

このネットのトポロジーは、多重 RBF ネットのそれと基本的に同一であると言える。しかし、GRNN は学習データから直接に関数の見積もりを引き出すため、入力データと出力データとの間に任意の関数を近似でき、関数予測に対して十分に確立された統計的手法

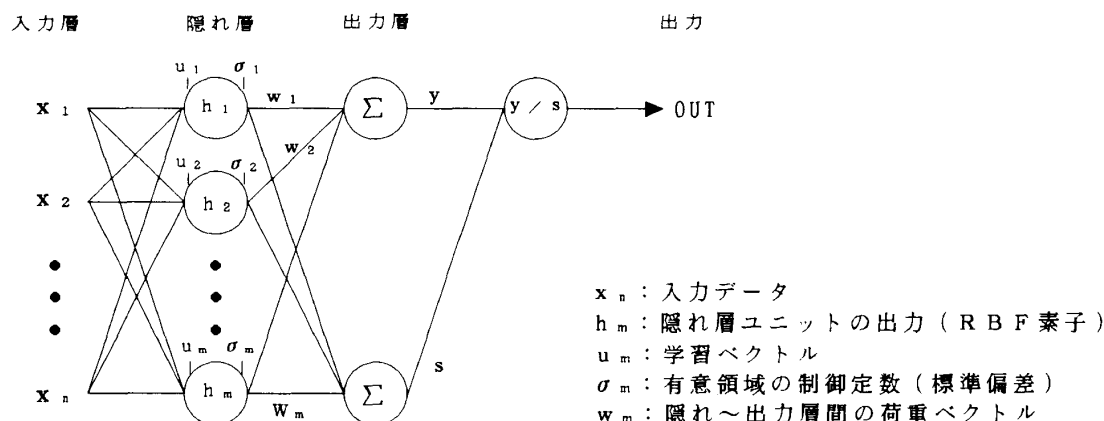


図4 GRNN の構成

であると言える。GRNN は、以下のような統計量に基づいている。

$$E[y|x] = \left\{ \int_{-\infty}^{\infty} yf(x, y) dy \right\} / \left\{ \int_{-\infty}^{\infty} f(x, y) dy \right\} \quad (15)$$

ここで、 x はネットに対する入力ベクトル、 y は出力である。さらに、 $E(x|y)$ は入力ベクトル x が与えられたときの出力の期待値であり、 $f(x, y)$ は x と y の同時確率密度関数 (PDF) を示している。PDF は、近似関数においての予備的な概念を何も無いデータから導くため、そのシステムは完全に一般的である。よって、GRNN は本質的に学習集合のみが与えられたときの $f(x, y)$ という関数を予測する方式であると言える。

以下に、GRNN における処理手順を記す。

【GRNN の処理手順】

Step 1: 標準偏差 σ , 隠れ層の中心位置 u , 荷重係数 w , 隠れ層のユニット数 h の決定 (前処理)

Step 2: 各入力データ X_m を全ての隠れ層ユニットに適用する。

Step 3: 各隠れ層ユニットにおいて、(11), (12) 式を用いて処理を行う。

Step 4: 各隠れ層ユニットは、荷重係数 w により出力層と結合され、以下に示す処理によって線形荷重の和を求める。

$$y = \sum_{m=1}^m h_m w_m \quad (16)$$

Step 5: 隠れ層の働きを、入力パターンに対して正規直交形に近い形で表現するため、以下の処理により正規化操作を行う。

$$out = \sum_{m=1}^m h_m w_m / \sum_{m=1}^m h_m \quad (17)$$

4. 関数近似の実施要領

本研究は、BP 及び GRNN を用い sin 関数、パルス関数について波形近似を行うものである。各アルゴリズムの実装は C 言語で行い、計算機は SUN-4/5 を使用した。

また、目標関数となるパルス関数 $pls(x)$ については、

$$pls(x) = \begin{cases} 1 & (\pi/2 \leq x \leq 3\pi/2) \\ 0 & (0 \leq x \leq \pi/2, 3\pi/2 < x \leq 2\pi) \end{cases}$$

のように、閉区間を設定した。以下に、各ニューラルネットの実施要領を記す。

【BP を用いたニューラルネット】

- 入力データ数：630 (0 ~ 2π 範囲 0.01刻み)
- 学習回数上限：5000回
- 学習係数 η の値：1.2
- 隠れ層の数：1層, 2層, 3層
- シグモイド関数の傾き ε ：1.0
- 各隠れ層におけるユニット数：5, 6, 7

とパラメータ及びネットの構成を設定した。なお、入力データと出力データとを実数の写像として扱うため、入力層及び出力層のユニット数は1に設定した。また、本実験におけるBPは、入出力関数にシグモイド関数を用いているため、教師信号とする各関数(sin関数, パルス関数)は正規化 ($0 \leq x < 1$, $0 \leq y \leq 1$) を行い、各パターンにおいて30回ずつ試行した。

【RBF を用いた GRNN】

- 入力データ数：630 (0 ~ 2π の範囲で0.01刻み)
- 標準偏差 σ^2 の値：1.5 ~ 0.00005 (11パターン)
- 隠れ層のユニット数 h ：1 ~ 10000 (20パターン)
- 隠れ層の中心位置 u の値：0 ~ 2π 内で隠れ層ユニット数に合わせて等間隔に設定
- 荷重係数 w の値：設定した u の値に対応した望ましい出力を設定

とパラメータを設定し、各パターンにおいて試行した。

なお、近似効率性を考える上で指標となる近似尺度は二乗誤差を使用した。

5. 実験結果及び考察

5.1 BP による関数近似

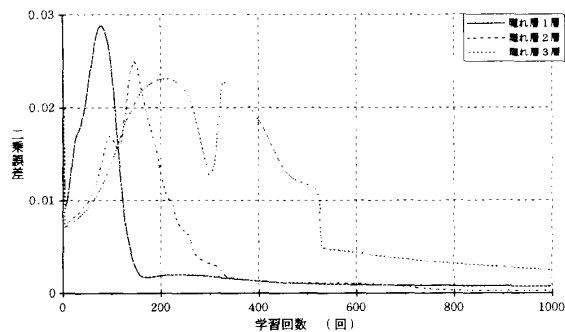
BP による sin 関数, パルス関数近似において、隠れ層数：1, 2, 3層および隠れ層のユニット数：5, 6, 7の全パターンについて実験を行った。グラフ1, 2は sin 関数, グラフ3, 4はパルス関数における隠れ層数別, ユニット数別の収束状況を示している。また、表1は、sin 関数近似における各学習回数に対する二乗誤差, 平均処理時間を示している。

以上の結果より、収束の点においては明らかに隠れ層数及びユニット数が増えるほど収束は遅くなっており、かなりの振動が見られる。sin 関数近似において、隠れ層を1層とした場合は少ない学習回数で収束が落ち着くが、それほど精度の高い近似値が得られていない。隠れ層2層では、収束は学習回数500回以内で落ち着き、それ以降もかなりの精度まで近似が行われている。しかし、隠れ層3層では学習回数が1500回以降かなりの収束が見ら

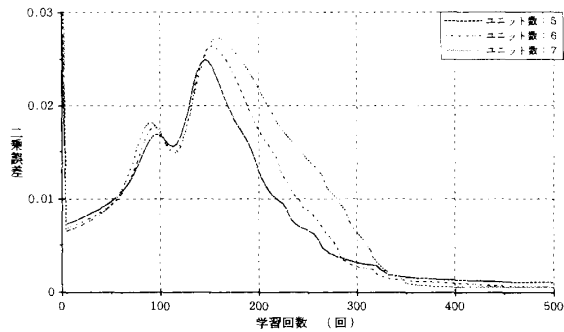
れるが、この収束は確率的なものであり、ユニット数が5(表1,*1):59.3%, 6(表1,*2):51.7%, 7(表1,*3):27.8%と増えるほど収束を行う割合が減少する。これはネットを多層にしたため、学習則の決定が難しくなったためだと考えられる。また、パルス波形近似においても隠れ層数別、ユニット数別の近似状況は、同様の傾向が表れている。

また、表1より処理時間の点において、隠れ層2層の場合は1層の約2.5倍ほど要するが、時間的にも2層の方が効率良く収束を行っている。また、隠れ層3層の場合は1層の約4倍、2層の約1.5倍の時間を要しており、時間的な収束は良いとは言えない。

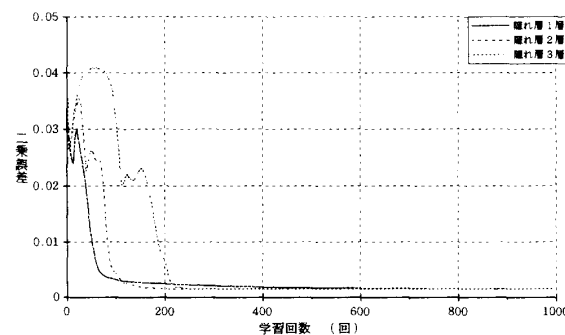
関数近似では目的関数・入力データを実数として扱うため、隠れ層数・ユニット数が増加するにつれ、BP アルゴリズムの特性上それに対する最適な荷重の組み合わせに、より学習回数と時間が要すると考えられる。今回の実験により、関数近似に対する最適な BP ネットは、隠れ層を2層、ユニット数を5個と設定することが望ましいと言える。



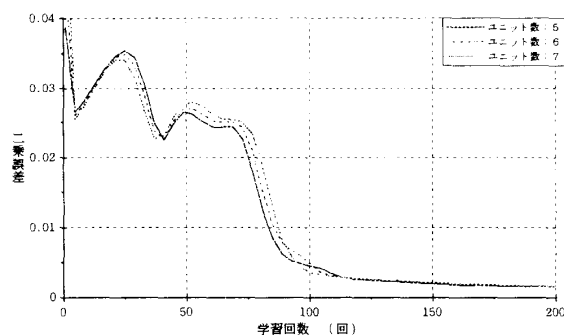
グラフ1 sin 関数近似における収束状況
(隠れ層: 1, 2, 3 層, ユニット数: 5)



グラフ2 sin 関数近似における収束状況
(隠れ層: 2 層, ユニット数: 5, 6, 7)



グラフ3 パルス関数近似における収束状況
(隠れ層: 1, 2, 3 層, ユニット数: 5)



グラフ4 パルス関数近似における収束状況
(隠れ層: 2 層, ユニット数: 5, 6, 7)

表1 sin 関数近似における学習回数別収束状況

学習回数(回)		500		1000		1500		2000		2500		3000	
隠れ層数	ユニット数	二乗誤差	平均処理時間(秒)	二乗誤差	平均処理時間(秒)	二乗誤差	平均処理時間(秒)	二乗誤差	平均処理時間(秒)	二乗誤差	平均処理時間(秒)	二乗誤差	平均処理時間(秒)
1 層	5	0.001056	37.433	0.000774	75.238	0.000614	112.977	0.000513	152.489	0.000439	190.897	0.000397	228.430
	6	0.001089	43.729	0.000776	87.562	0.000639	131.326	0.000536	174.372	0.000451	218.525	0.000404	262.461
	7	0.001124	49.257	0.000781	99.756	0.000649	148.894	0.000545	198.762	0.000463	248.014	0.000412	298.569
2 層	5	0.000964	90.547	0.000260	181.398	0.000160	273.647	0.000131	364.863	0.000116	455.343	0.000104	547.329
	6	0.000894	115.285	0.000258	230.750	0.000183	345.214	0.000148	460.285	0.000122	574.167	0.000108	689.969
	7	0.000742	142.699	0.000245	284.175	0.000189	425.398	0.000152	567.197	0.000127	709.231	0.000111	851.211
3 層	5	0.013427	144.361	0.002872	288.539	0.000205	432.057	0.000124	577.210	0.000093	723.853	0.000076	868.524 *1
	6	0.019228	186.622	0.001789	373.664	0.000187	561.477	0.000129	748.638	0.000104	935.328	0.000096	1122.438 *2
	7	0.019682	235.341	0.001494	470.752	0.000182	707.250	0.000138	943.902	0.000116	1180.321	0.000109	1417.265 *3

5.2 RBF を用いた GRNN による関数近似

RBF を用いた GRNN による \sin 関数、パルス関数近似において、標準偏差 σ^2 の値を 11 種類、隠れ層のユニット数を 20 種類取り上げ、すべてのパターンについて実験を行った。グラフ 5, 6 は各々 \sin 関数、パルス関数近似における各ユニット数に対する標準偏差 σ^2 の値による二乗誤差の推移を示したものである。グラフ 7 は各ユニット数に対する平均処理時間を示したものである。また、グラフ 8, 9 は GRNN による \sin 近似波形の一例を示す。

有効となる入力データを考慮するユニットの有意領域は、 σ 値により決定される領域の直径により、システムの正確さに深く影響を与える。よって、 σ を決定する目的は、出来るだけ均等に有意領域をカバー出来るようにすることであると言える。また、入力データが中心 u よりも、かなり大きく離れている隠れ層ユニットに対しては十分に大きな σ の値をとる必要があると言える。グラフ 5, 6 より、 σ の値を小さくすれば近似度が高い傾向にあるが、ユニット数によっては $\sigma^2:0.005$ を境に近似が悪くなっている。これは、ユニット数に対する σ の値が小さすぎたためであると思われる。よって有効な σ^2 値は近似する関数によって異なり、またユニット数によっても異なってくると言える。

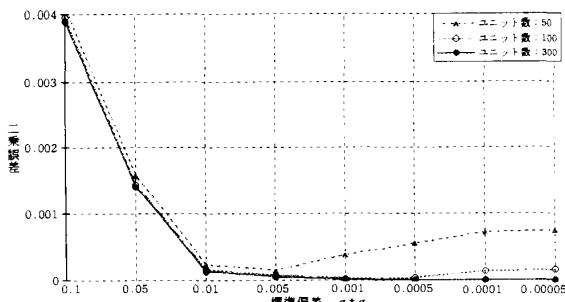
また、グラフ 8, 9 よりユニット数の違いにより近似の仕方が異なっていることが言える。グラフ 8 に示すような階段状の波形は、ユニット数が少ないため、ターゲットとして与えた荷重の値を大きく反映したためだと思われる。よって、階段の数はユニット数と一致することが言える。また、 2π 付近の \sin 近似が良くないのは、その周辺における荷重係数 w を考慮していないため波形が流れていると言える。

\sin 関数近似においてはグラフ 5 より、隠れ層のユニット数を増やすほど近似度は良くなると言える。しかし、グラフ 7 に示すように、ユニット数を増やすにつれて処理時間がかかるという欠点が生じる。また、ユニット数を増やせばより良い近似は得られるが、ユニット数が 300 以降は目に見える違いは表れず、むしろ σ^2 の値によって異なることが言える。パルス関数近似においても、グラフ 6 より局所的にユニット数:50 で良い近似が行われているが同様な傾向が見られる。

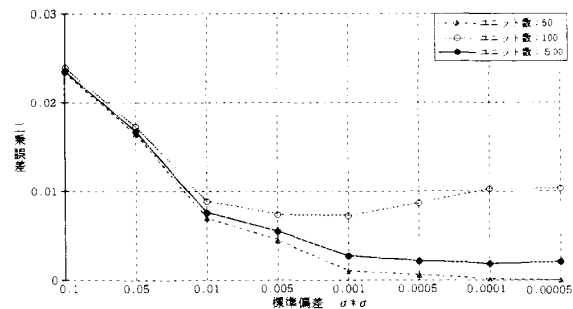
以上の結果より、今回の実験で行った近似波形の最適な標準偏差 σ^2 の値、隠れ層ユニットの数を以下の表 2 に記す。

表 2 RBF による各関数近似における最適な諸設定

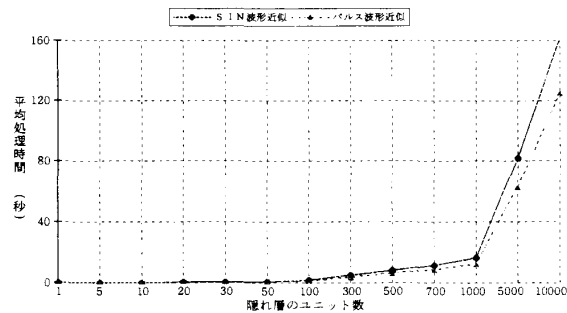
	標準偏差 σ^2 の値	隠れ層のユニット数
\sin 関数	0.001 前後	300 程度
パルス関数	0.0005 前後	50 程度



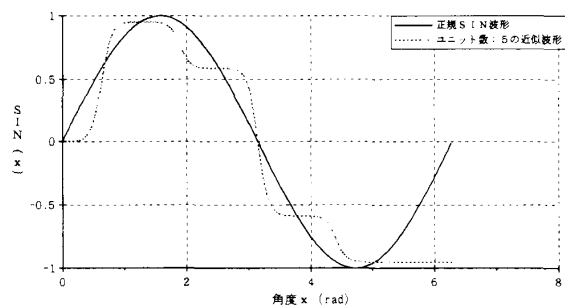
グラフ 5 隠れ層のユニット数別 二乗誤差 (sin 関数近似)



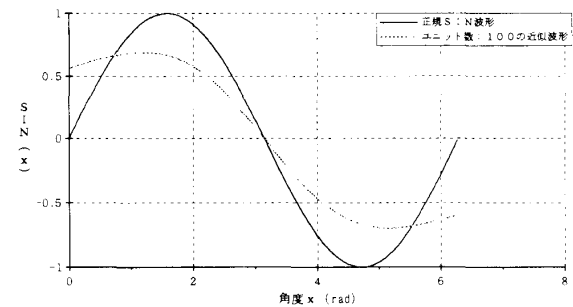
グラフ 6 隠れ層のユニット数別 二乗誤差 (パルス関数近似)



グラフ 7 隠れ層のユニット数別 平均処理時間



グラフ 8 GRNN による sin 近似波形 (ユニット数: 5, 標準偏差 σ^2 : 0.1)



グラフ 9 GRNN による sin 近似波形 (ユニット数: 100, 標準偏差 σ^2 : 1.0)

5.3 関数近似における BP と GRNN

次に、関数近似における BP と RBF を用いた GRNN の比較を行う。BP については、隠れ層: 1, 2, 3 層, ユニット数: 5, 学習回数上限: 5000 回と設定を行い, また GRNN については、隠れ層のユニット数を 10~100 までの 7 種類, 標準偏差 σ^2 の値を 0.05~0.0001 までの 4 種類ほど取り上げネットを設定し, 各パターンについて実験を行った。

グラフ 10 (a)~(d) は BP, グラフ 11 (a)~(d) は GRNN を用いた場合で, それぞれ二乗誤差が 0.01, 0.005, 0.001, 0.0001 における sin 近似波形を示す。同様にグラフ 12 (a)~(d), グラフ 13 (a)~(d) は, BP, GRNN による二乗誤差を 0.03, 0.01, 0.005, 0.001 におけるパルス近似波形を示す。また表 3, 4 は各々 sin 関数, パルス関数近似における BP, GRNN の平均処理時間を記したものである。

表3 BP, GRNN による sin 関数近似における処理時間

BP 学習			二 乗 誤 差								
隠れ層数 (ユニット数: 5)			0.01	0.007	0.005	0.003	0.001	0.0007	0.0005	0.0003	0.0001
1 層	平均処理時間(秒)		0.155	10.019	10.019	10.881	39.161	95.379	164.909	369.258	*
	平均学習回数(回)		2.0	129.9	131.5	141.1	506.0	1231.9	2127.6	4762.8	*
2 層	平均処理時間(秒)		0.369	44.660	45.063	55.661	68.520	81.735	89.788	117.267	410.173
	平均学習回数(回)		2.0	245.3	247.6	305.1	375.3	446.9	490.9	639.9	2009.2
3 層	平均処理時間(秒)		0.579	55.816	257.771	269.791	353.328	467.974	502.028	504.525	703.120
	平均学習回数(回)		2.0	192.0	888.7	929.9	1216.6	1609.7	1718.1	1732.1	2403.9
R B F		平均処理時間(秒)	0.320	0.343	0.447	0.480	0.500	0.590	0.730	0.952	1.790

表4 BP, GRNN によるパルス関数近似における処理時間

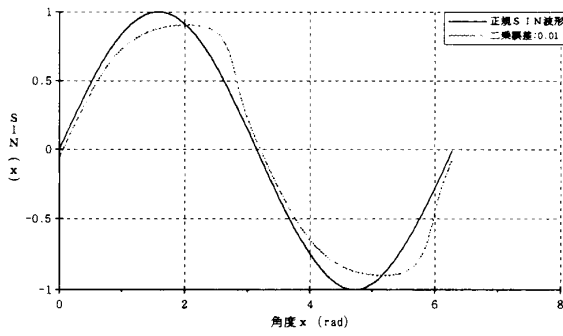
BP 学習		二 乗 誤 差						
隠れ層数 (ユニット数: 5)		0.03	0.01	0.007	0.005	0.003	0.002	0.001
1 層	平均処理時間(秒)	0.155	3.845	4.080	4.679	8.671	28.376	*1
	平均学習回数(回)	2.0	49.9	52.9	60.7	112.0	364.1	*1
2 層	平均処理時間(秒)	0.367	14.933	15.696	15.891	18.894	26.029	*2
	平均学習回数(回)	2.0	81.6	85.7	87.0	103.4	142.1	*2
3 層	平均処理時間(秒)	0.583	52.567	53.844	55.294	55.444	64.049	1047.626
	平均学習回数(回)	2.0	180.94	185.4	190.3	190.8	220.2	3557.1
R B F	平均処理時間(秒)	0.275	0.916	1.160	1.860	3.018	6.360	8.862

表5 GRNN による関数近似における例外的な処理時間

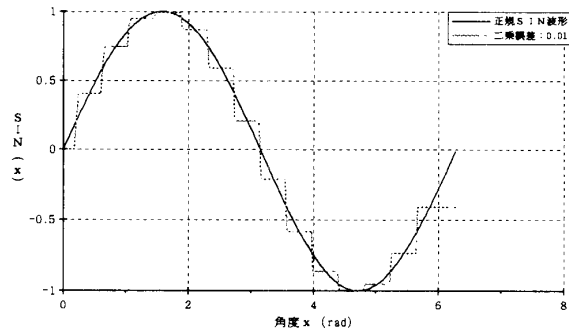
	二乗誤差	ユニット数	標準偏差値	処理時間(秒)
sin 関数近似	0.1	1000	1.5	16.410
	0.03	5000	0.5	83.460
	0.001	10000	0.01	170.870
パルス関数近似	0.02	800	0.0005	10.210
	0.005	10000	0.005	126.470
	0.001	10000	0.0001	127.830

以上の結果より, BP 及び GRNN を用いた場合の目標関数に対する近似過程の相違点が明らかに見られる。BP の近似波形は, 全体的に歪んでおり学習が進むにつれ徐々に修正を行っている。しかし y 軸の範囲が, sin 関数においては 0.5 以上及び -0.5 以下, パルス関数においては 0.8 以上及び 0.2 以下の近似が局所的にうまく行われていないと言える。これは, 今回の BP において, 目標となる関数(教師信号)に対して正規化を行い, 入力データ及び出力データを実数として扱ったのだが, 入出力関数にシグモイド関数を使用しているため, “1”, “0” への収束が非常に困難であるためだと考えられる。しかし, 先に記した範囲以外の近似は, GRNN より正確に行われている。

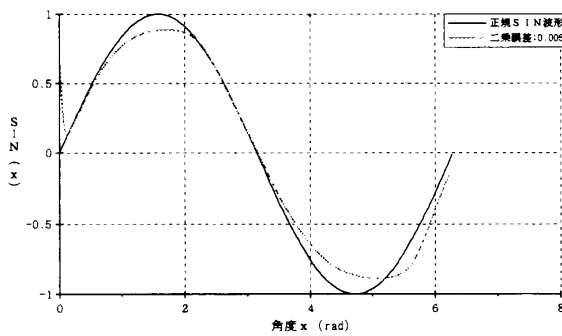
これに対して GRNN による近似波形は, 全体的に均衡のとれた形をしている。グラフ 11 (a)~(d) の sin 関数近似においては, 隠れ層のユニット数を 100 以下に設定しているため, 階段状の波形になっているが, ユニット数を増やし標準偏差 σ^2 の値を小さくするにつれ近似が正確に行われている。



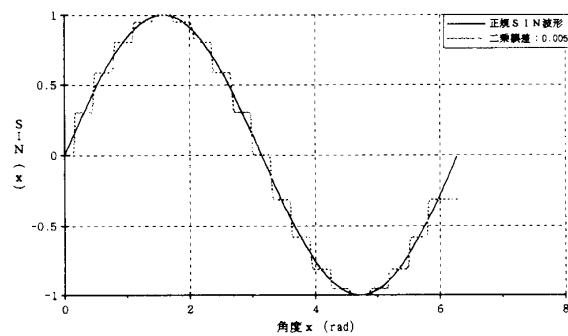
グラフ10(a) BPによるsin関数近似
二乗誤差: 0.01



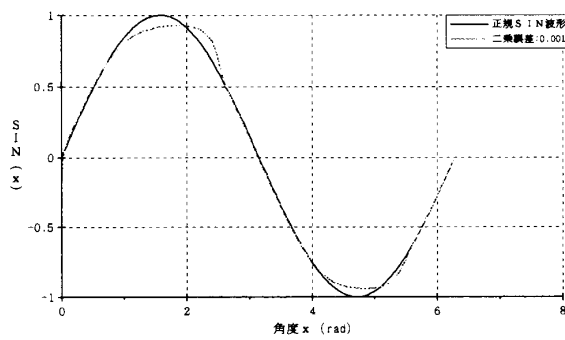
グラフ11(a) GRNNによるsin関数近似
二乗誤差: 0.01 (ユニット数15, $\sigma^2: 0.001$)



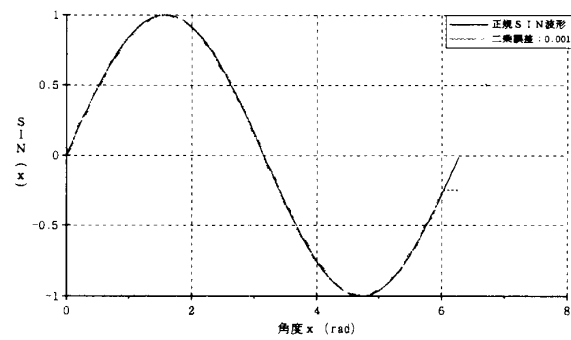
グラフ10(b) BPによるsin関数近似
二乗誤差: 0.005



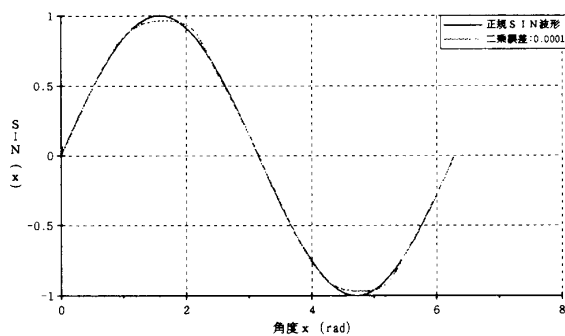
グラフ11(b) GRNNによるsin関数近似
二乗誤差: 0.005 (ユニット数: 20, $\sigma^2: 0.001$)



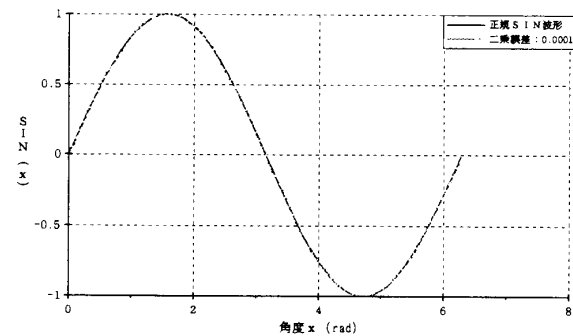
グラフ10(c) BPによるsin関数近似
二乗誤差: 0.001



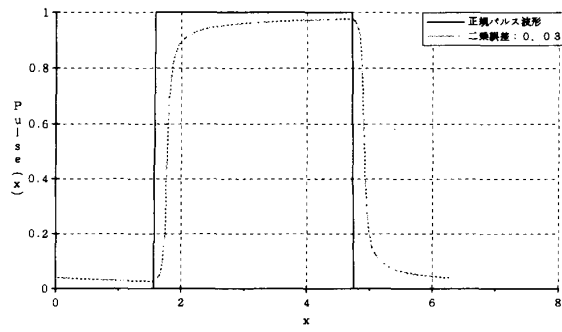
グラフ11(c) GRNNによるsin関数近似
二乗誤差: 0.001 (ユニット数: 25, $\sigma^2: 0.01$)



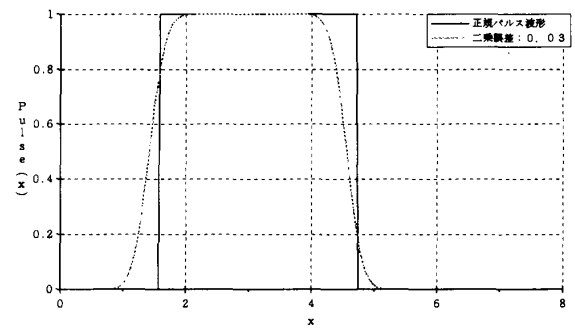
グラフ10(d) BPによるsin関数近似
二乗誤差: 0.0001



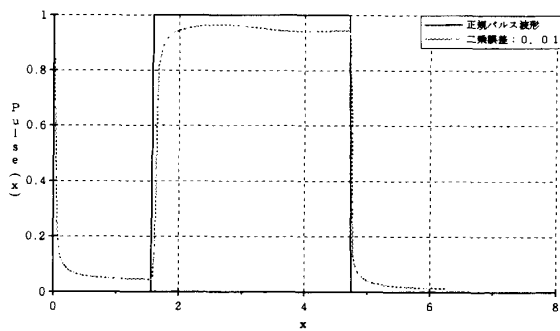
グラフ11(d) GRNNによるsin関数近似
二乗誤差: 0.0001 (ユニット数: 100, $\sigma^2: 0.0001$)



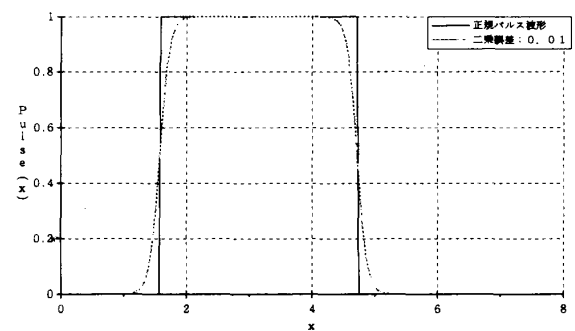
グラフ12(a) BPによるパルス関数近似
二乗誤差：0.03



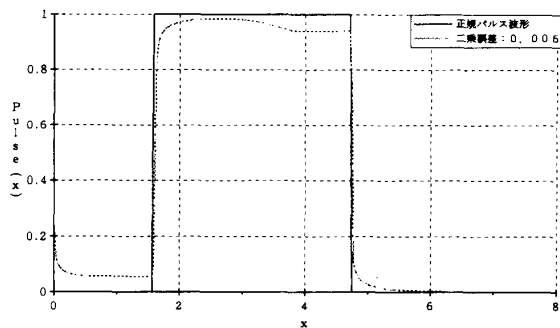
グラフ13(a) GRNNによるパルス関数近似
二乗誤差：0.03 (ユニット数：20, σ^2 ：0.05)



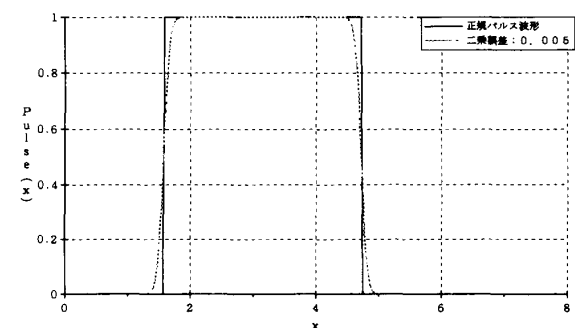
グラフ12(b) BPによるパルス関数近似
二乗誤差：0.01



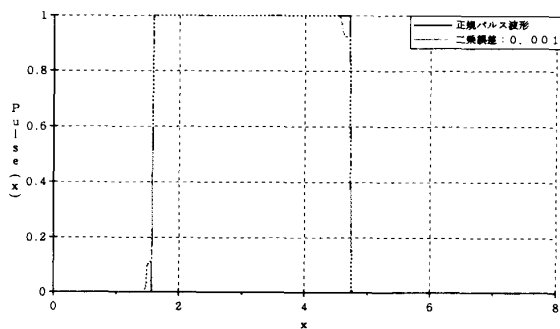
グラフ13(b) GRNNによるパルス関数近似
二乗誤差：0.01 (ユニット数：10, σ^2 ：0.005)



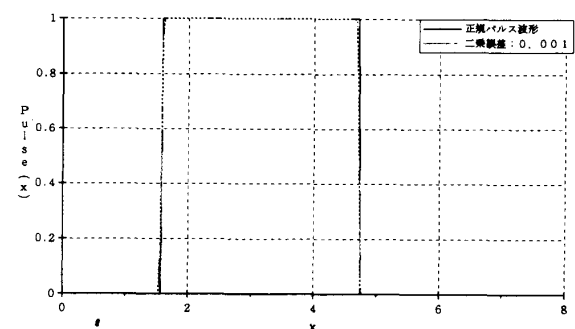
グラフ12(c) BPによるパルス関数近似
二乗誤差：0.005



グラフ13(c) GRNNによるパルス関数近似
二乗誤差：0.005 (ユニット数：30, σ^2 ：0.01)



グラフ12(d) BPによるパルス関数近似
二乗誤差：0.001



グラフ13(d) GRNNによるパルス関数近似
二乗誤差：0.001 (ユニット数：50, σ^2 ：0.001)

また、処理時間の点においても表3, 4より、BPによる関数近似においては、隠れ層2層の場合が最も最適であるという結果が得られているが、GRNNと比較するとsin関数近似においては約100~500倍、パルス関数近似においては約4~10倍ほど時間がかかっていることが分かる。よって処理時間的に比較すると、明らかにGRNNによる近似の方が早いことが言える。しかし、表5に記すようにGRNNによるsin関数近似において、隠れ層のユニット数を10000、標準偏差 σ^2 を0.001と設定し、波形を二乗誤差:0.001に近似させた場合、その処理時間はBPにおける隠れ層1, 2層の場合の約3~4倍の170秒ほどかかるという例外的な結果が得られた。これにより、GRNNによる手法はユニット数、標準偏差値の設定により、BPによる近似よりも処理時間を要する場合があると言える。なお、表3における(*)欄、表4における(*1), (*2)欄は、BPによる近似において学習回数:5000回では、設定した二乗誤差に収束出来なかったことを意味するものである。

以上より関数近似においては、近似波形及び処理時間を比較検討を行った結果、BPよりGRNNの方がより適するものであると言える。

6. おわりに

本研究では、ニューラルネットにおける関数近似としてBPとRBFを用いたGRNNを取り上げ、各手法における特性及び両者の比較による有効性を検討した。

BPについては、学習係数やシグモイド関数の傾き等のパラメータを一定に設定し、隠れ層数及びユニット数を変化させ、その場合における近似度、処理時間の算出を行った。本来、ニューラルネットは多層化する事により処理能力が向上するとされているが、多層ニューラルネットワークにおける学習則を見つけることは非常に困難なため、今回の実験では隠れ層数は2層が最適であるという結果が得られた。また、一般にBPネットにおけるパラメータ等の諸設定はある程度容易であり、目標とする関数を正確に設定すれば、どのような入力データの分布に対してもある程度の近似は行え、様々な応用が可能になると考えられる。しかし、処理時間の点においてはかなりの時間を必要とし、効率的にはあまり良い手法であるとは言えない。

これに対しGRNNにおいては、最適な隠れ層数、ユニット数の設定が近似させたい関数によって異なり、関数に対する近似度とそれに対する処理時間との関係の考慮が、重要かつ必要であると言える。また今回、近似させたい波形及び入力データの分布が把握出来ていたため、隠れ層の中心位置 u 、荷重係数 w 等の設定が容易であった。しかし、入力データ分布が全く未知の場合、入力パターン空間に均等になるように u の値を設定しなければならない。この様に、様々なパラメータに対する設定、つまり前処理の複雑な点がGRNNの欠点であると言える。しかし、諸設定を上手に行えば、近似も極めて正確に行うことができ、また処理時間も数秒以内であるため、関数近似においてはBPよりもGRNNの方が大変有効であると言える。

現在, ニューラルネットの一分野として, BP と同様にパターンの分類を行うカウンタープロパゲーション (Counter Propagation: CP) が提唱されているが, 今後, CP を用いた関数近似における特性, 有効性の検証及び BP や GRNN との比較検討を行う予定である。

参考文献

- 1) R.ヘクト・ニールセン:「ニューロコンピューティング」, トッパン (1992).
- 2) 久間和生・中山 高:「ニューロコンピュータ工学」, 工業調査会 (1992).
- 3) R.ピール・T.ジャクソン:「ニューラルコンピューティング入門」, 海文堂 (1993).

Function Approximation by Neural Networks

Hiroiyuki NARIHISA and Noriaki FUKUDA*

Department of Information & Computer Engineering

**Graduate School of Engineering*

Okayama University of Science,

Ridai-cho 1-1, Okayama 700, Japan

(Received October 7, 1996)

The current interest in artificial Neural Networks is largely a result of their ability to mimic natural intelligence.

As a result of this research, artificial Neural Networks have been used in a broad range of applications. These include pattern classification, pattern completion, function approximation, optimization, prediction, and automatic control.

In essence, they accept a set of inputs and produce a corresponding set of outputs. Therefore, they may be called a kind of mapping.

In this paper, we represent a behavior characteristic of Neural Networks by confining the function approximation.